Overview

CPSC 4470/5470

Introduction to Quantum Computing

Instructor: Prof. **Yongshan Ding**

Computer Science, Applied Physics, Yale Quantum Institute

Our Modern Digital World

We use binary digits to store, process, and communicate information

Binary Digits	Data	Software	Models
Series of 0s and 1s.	Numbers, text, images, videos, internet, social media, neural networks, intelligent models …		

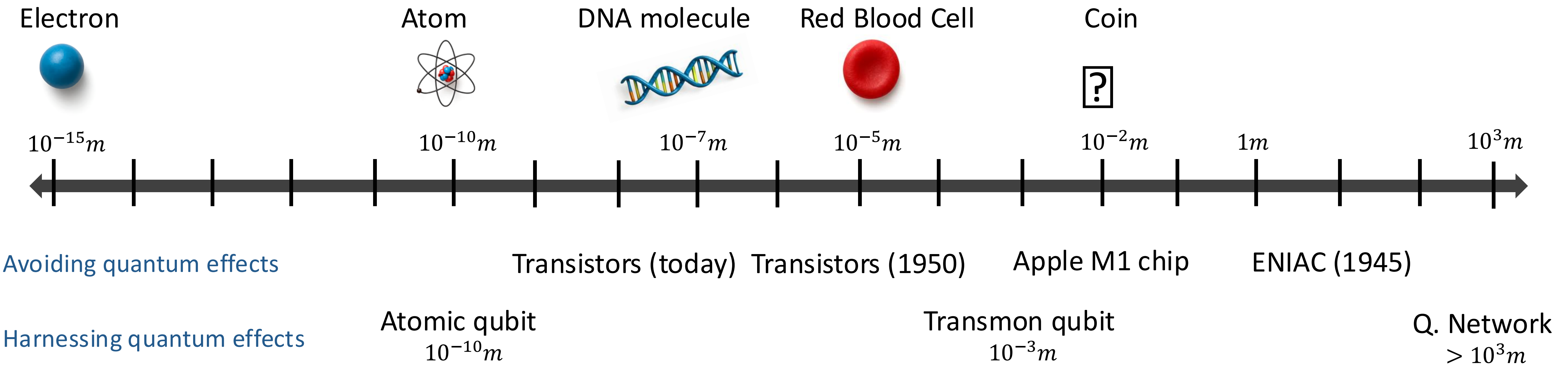
Binary Information

Unit	Bit: $b \in \{0,1\}$
Possible State	0 or 1
Physical Carrier	**Transistor**: on/off **Voltage in wire**: high/low

Image generated from ChatGPT.

Nature’s Language: Quantum Mechanics

Microscopic: Must use quantum mechanics to describe. **Macroscopic**: Can often ignore the effects of quantum mechanics.



Qubits: Accessing Quantum Properties

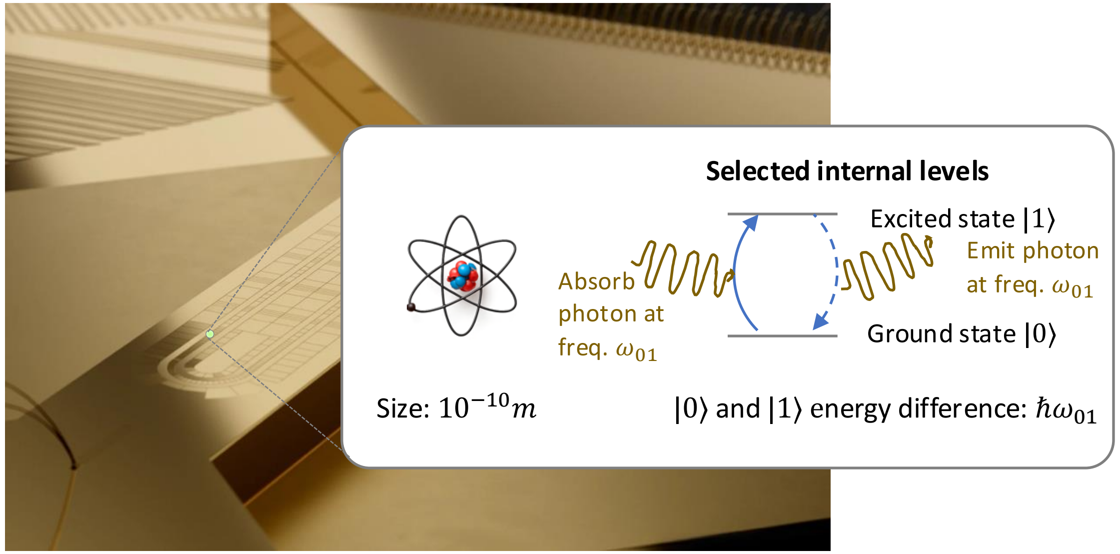
Quantum Information

Qubit: $|\psi\rangle \in \mathbb{C}^2$

$|\psi\rangle = 0.6|0\rangle + 0.8|1\rangle$ ← Dirac notation

Atoms: internal energy levels, **Photons**: polarizations, **Superconducting circuits**: Persistent current

Natural atom: **Trapped Ions/Rydberg Atoms**



Artificial atom: **Superconducting Circuits**

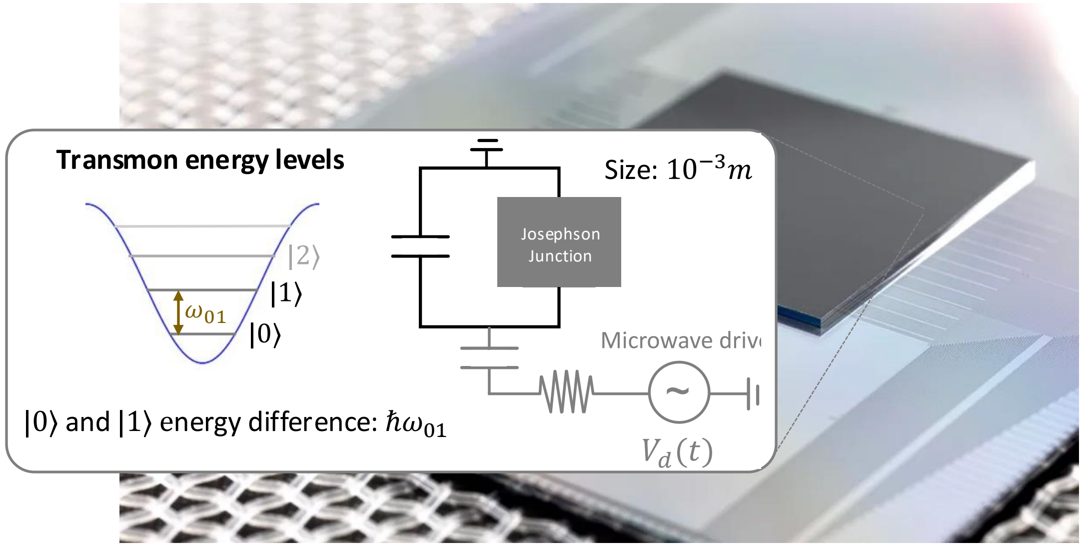
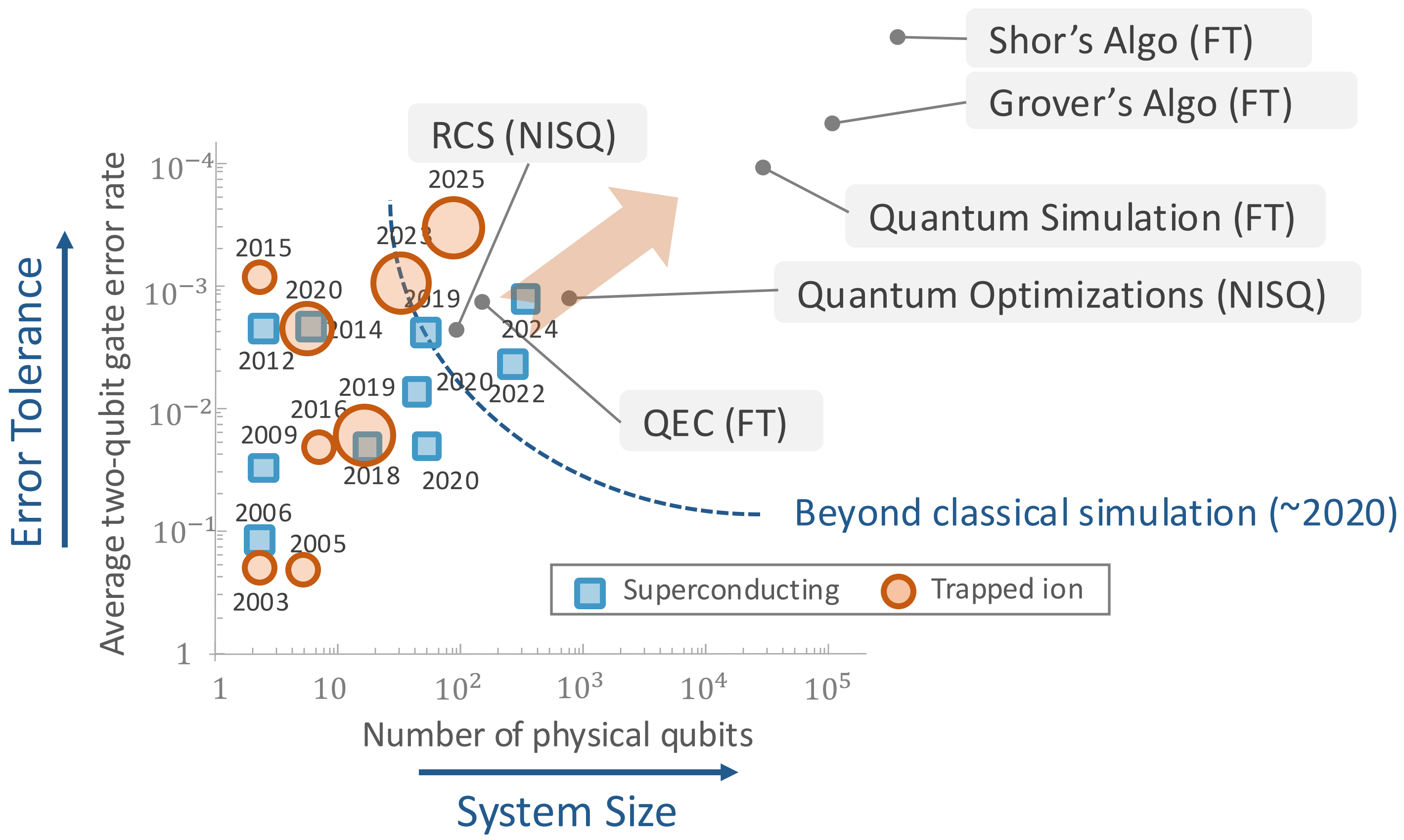


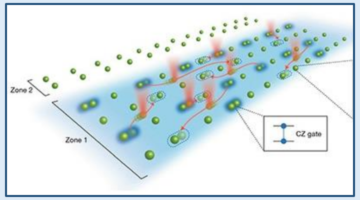
Image credit: Quantinuum (left), Google Quantum AI (right).

Hardware Improvements Over the Years

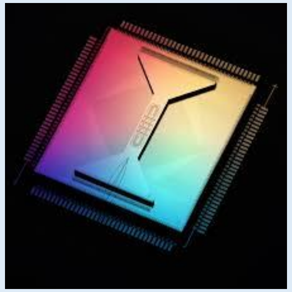


*Size of data point indicates connectivity; larger means denser connectivity.

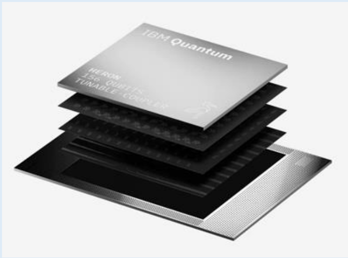
Progress in 2024-25



Rydberg Atom Arrays
256 qubits [QuEra]
99.5% gate fidelity
Atom movements



Trapped Ions
56 qubits [Quantinuum]
99.9% gate fidelity
All-to-All Connectivity
Mid-circuit measurements



Superconducting Circuits
156 qubits [IBM]
99.9% gate fidelity
Mid-circuit measurements

Sources (from left to right, then top to bottom): Ding & Chong, Harvard/QuEra, Quantinuum, IBM

Early Computers – Filling Up an Entire Room

IBM System/360 at NASA (1960s)

IBM Quantum (2019)

Google's Willow Chip (2024)

It's going to be a challenging journey before we build functional quantum computers.

Fortunately, this time around, we can use powerful digital computers to help us.

Photos from: en.wikipedia.org, IBM Quantum, Google Quantum AI.

Solve Problems Faster with a Quantum Computer

Some problems are **easy**, in terms of resources in space (memory) or time (steps):

- Multiplying two numbers
  - Long multiplication: time complexity $O(n^2)$,for n-digit numbers.
  - Schönhage-Strassen algorithm (1968): time complexity $O(n \log n \log \log n)$

Some problems are **hard:**

- Factoring a 2048-bits long number (RSA-2048): no poly(n)-time algorithm is known.

But they might be **easier in a quantum world:**

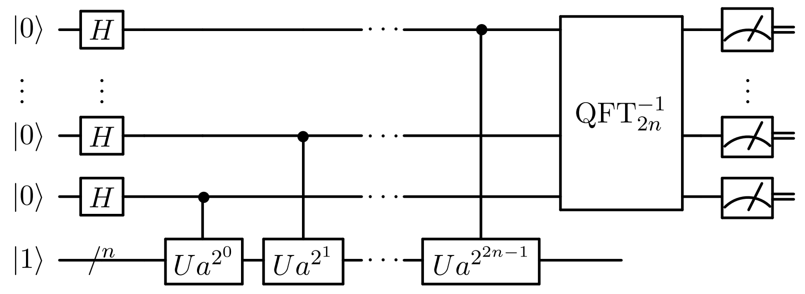
- Shor's factoring algorithm (1994): $O(n^2 \log n)$ elementary quantum operations
- Hint: Given 1000 qubits, their joint state is described by 2^1000 (complex) numbers.

Solve Problems Faster with a Quantum Computer

Prime Factorization [Shor, 1994]

1. Pick a random number $1 < a < N$.
2. Compute $K = \gcd(a, N)$, the greatest common divisor of a and N.
3. If $K \neq 1$, then K is a nontrivial factor of N, with the other factor being $\frac{N}{K}$ and we are done.
4. Otherwise, use the quantum subroutine to find the order r of a.
5. If r is odd, then go back to step 1.
6. Compute $g = \gcd(N, a^{r/2} + 1)$. If g is nontrivial, the other factor is $\frac{N}{g}$, and we're done. Otherwise, go back to step 1.

Quantum subroutine



We will learn more about Shor's algorithm in Lecture 16.

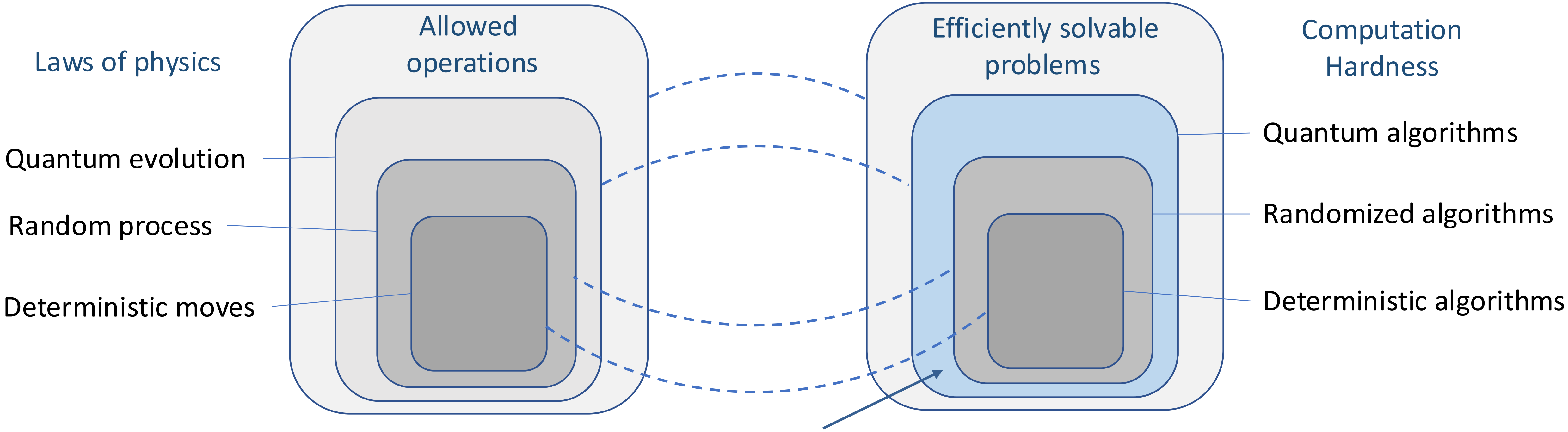
Source: https://en.wikipedia.org/wiki/Shor%27s_algorithm

The Physics of Computation

The first Conference on the Physics of Computation in May 1981 at MIT

Paul Benioff: A computer that operates under the law of quantum mechanics.
Richard Feynman: Simulating quantum systems needs a quantum computer.

The laws of physics determines what kinds of computation can be done (efficiently).



Finding problems that are classically hard but quantumly easy.

Where to Find Quantumly Easy Problems?

Some problems are hard to compute, in terms of resources in space (memory) and time (steps).
But easier in a quantum world.

Quantum Simulation [Feynman, 1982]

> "The full description of quantum mechanics for a large system with R particles… has too many variables, it cannot be simulated with a normal computer with a number of elements proportional to R…
>
> And therefore, the problem is, how can we simulate the quantum mechanics? … We can give up on our rule about what the computer was, we can say:
>
> **Let the computer itself be built of quantum mechanical elements which obey quantum mechanical laws.** "

There's a class of computational problems that are inherently quantum.

CPSC 4470/5470: Introduction to Quantum Computing

Quantum computational thinking – *how to use superposition and entanglement to solve problems.*

Instructor: Prof. Yongshan Ding
Course Website: https://www.yongshanding.com/cpsc447-f25/

MODULE 1	MODULE 2	MODULE 3	MODULE 4
Basic Quantum Info	Quantum Programming	Quantum Algorithms	Noisy Quantum Systems
> Quantum parallelism > Quantum entanglement > Gates and measurements	> Quantum circuits > Universal compilers > Classical vs quantum logic	> Quantum oracles > Quantum Fourier transform > Quantum factoring > Quantum searching	> Noisy states and channels > Error correcting codes > Quantum fault tolerance
WEEK 1 – WEEK 3	WEEK 4 – WEEK 5	WEEK 6 – WEEK 11	WEEK 12 – WEEK 15
		OCT RECESS (WEEK 8)	NOV RECESS (WEEK 14)

Pre-requisite: CPSC 2010 and CPSC 2020, or equivalents.
We will use the following **tools**: Canvas (for course materials), Gradescope (for HW/grades), Ed Discussions (for Q&A).

State of a Qubit

A qubit can be in a "superposition state" of 0 and 1 simultaneously.
The state is given by the following linear combination:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Where α and β are complex numbers: $|\alpha|^2 + |\beta|^2 = 1$.

When the qubit is measured, the state collapses to one of its basis states at random:

Example: $|\psi\rangle = 0.6|0\rangle + 0.8|1\rangle = \begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix}$

$$0.6^2 + 0.8^2 = 1$$

Measure:

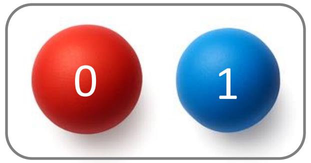
- 0 $p = |\alpha|^2 = 0.6^2$

or

- 1 $p = |\beta|^2 = 0.8^2$

Storing Data in Classical vs Quantum Register

Classical register
of 2 bits

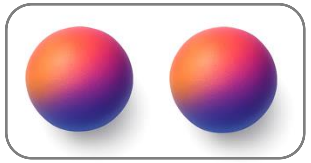


$x = 2$

x	binary
0	00
1	10
2	01
3	11

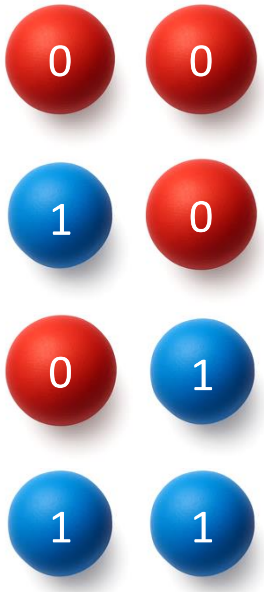
Note: Least significant bit writes at the leftmost index.

Quantum register
of 2 qubits



$$|\psi\rangle = 0.5|0\rangle + 0.5|1\rangle + 0.5|2\rangle + 0.5|3\rangle$$
$$= 0.5|00\rangle + 0.5|10\rangle + 0.5|01\rangle + 0.5|11\rangle$$

Readout/Measure:



$p = 0.25$

$p = 0.25$

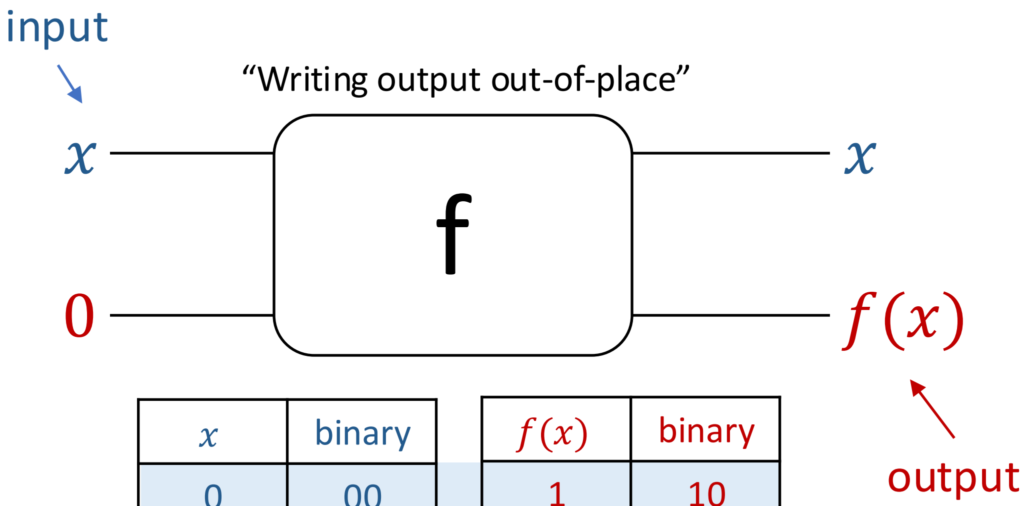
$p = 0.25$

$p = 0.25$

Abstract Computational Model

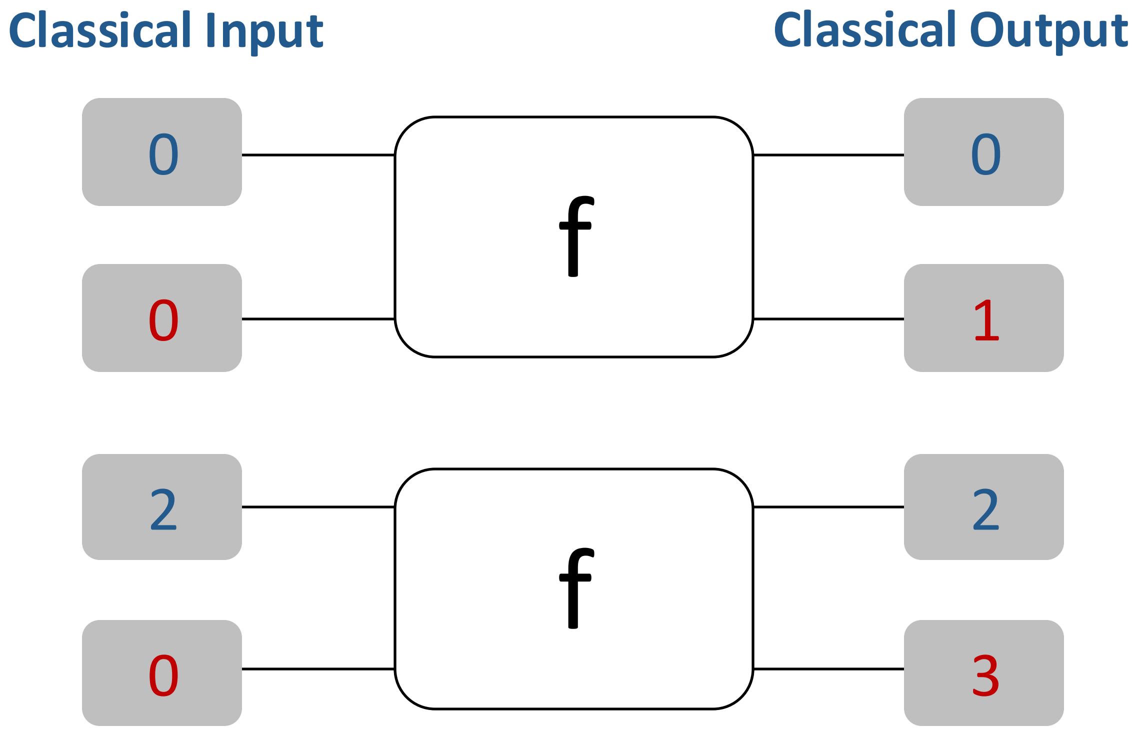
Circuit Model

Goal: compute function f(x) = x + 1 (mod 4)



x	binary
0	00
1	10
2	01
3	11

$f(x)$	binary
1	10
2	01
3	11
0	00



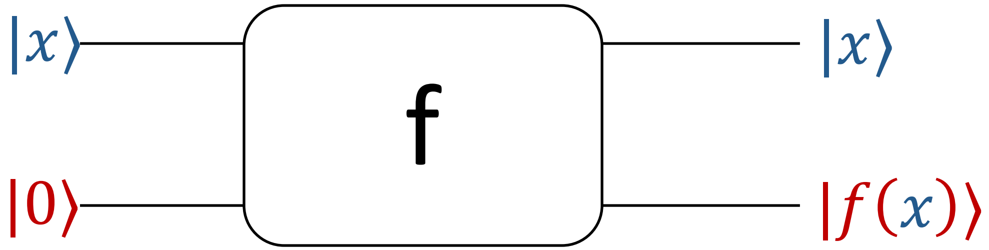
Evaluate $f(x)$ on one input x at a time.

Quantum Computational Model

Quantum Circuit Model

Goal: compute function f(x) = x + 1

Evaluate $f(x)$ on multiple inputs x in superposition.



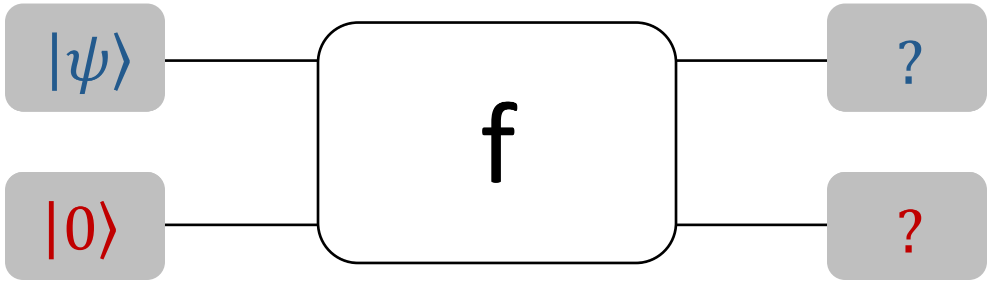
x	binary
0	00
1	10
2	01
3	11

$f(x)$	binary
1	10
2	01
3	11
0	00

Quantum Input

$(0.6|0\rangle + 0.8|2\rangle)|0\rangle$

Quantum Output



What should the output quantum state be?

Hint: think about the results we want when we measure the qubits.

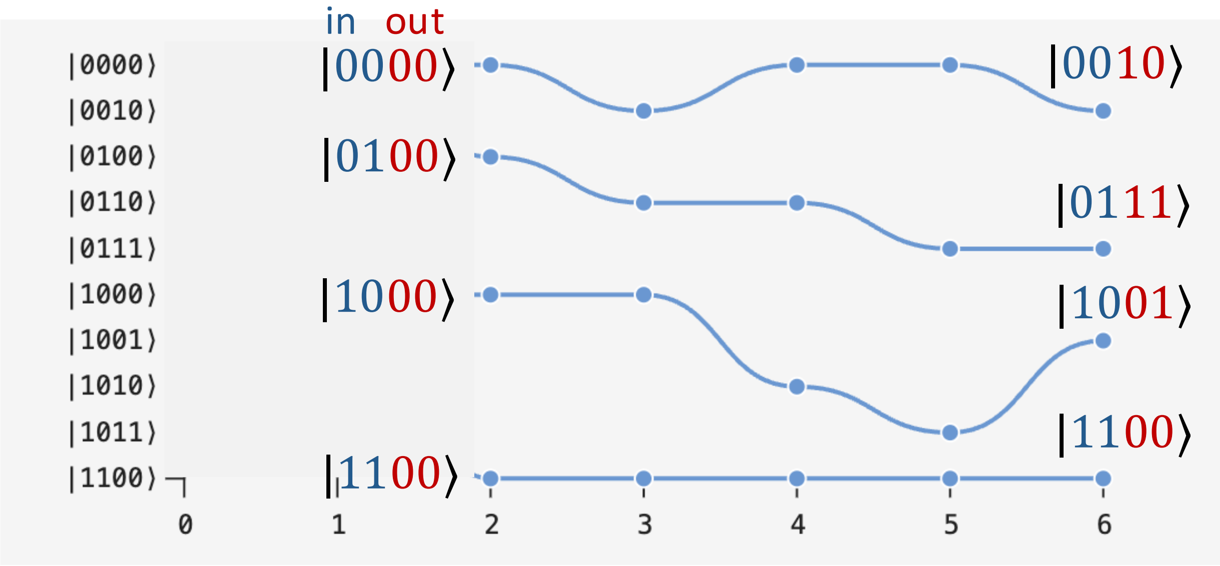
- $(0.6|0\rangle + 0.8|2\rangle)(0.6|1\rangle + 0.8|3\rangle)$?
- $(0.6|0\rangle|1\rangle + 0.8|2\rangle|3\rangle)$?

Need to use the “joint state” of two qubits to describe.

Quantum Computational Model

Visualizing Quantum Parallelism

Goal: compute function f(x) = x + 1



Evaluate $f(x)$ on multiple inputs x in superposition.

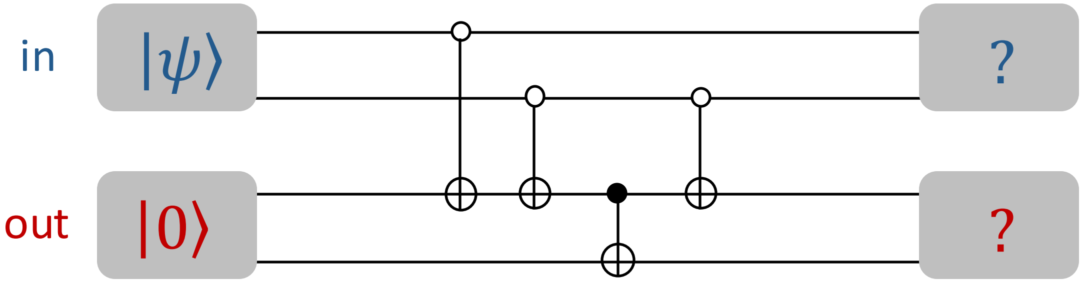
Is it simply evaluating $f(x)$ on multiple inputs x in parallel?

Quantum Input

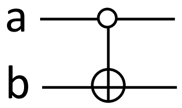
$(0.6|0\rangle + 0.8|2\rangle)|0\rangle$

Quantum Output

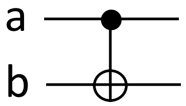
$0.6|0\rangle|1\rangle + 0.8|2\rangle|3\rangle$



Controlled-Not Gates:



If a=0, flip b; otherwise, do nothing.



If a=1, flip b; otherwise, do nothing.

Data Parallelism: Single Program Multiple Data

H100 GPU

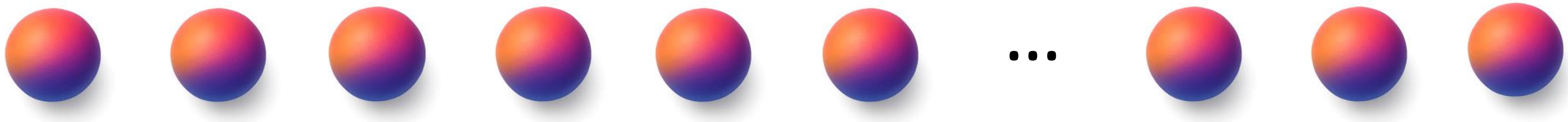
- 144 SM per chip
- 128 FP32 CUDA Cores per SM, 66.9 TFLOPS for FP32
- 2048 threads per SM (32 threads per warp, 64 warps per SM)
- 60MB L2 Cache, 80 GB GPU memory (HBM3)

- This chip can execute up to 294,912 CUDA threads!
- For high-density arithmetic and local data, this is highly efficient.

Image credit: NVIDIA

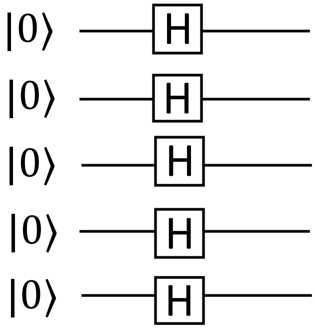
Superposition and Interference

The **massive** quantum parallelism

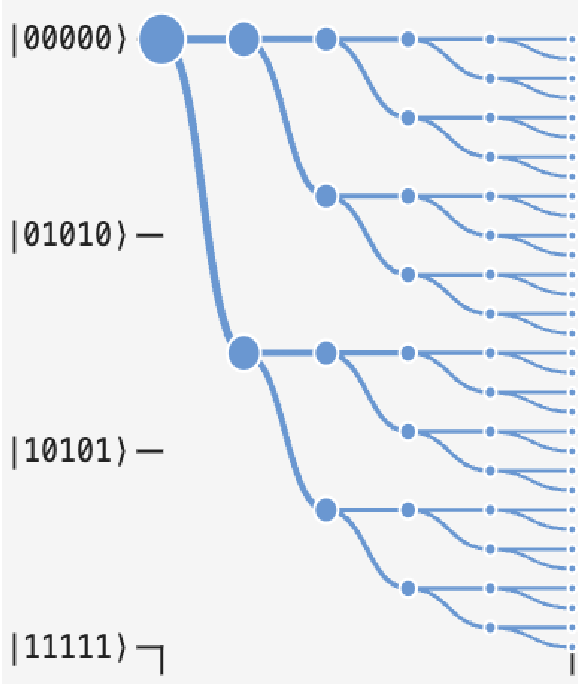


- For 10 qubits, writing down the joint state, we need:
    - $2^{10} \approx 10^{3}$ a thousand complex numbers
- For 20 qubits:
    - $2^{20} \approx 10^{6}$, a million complex numbers
- For 30 qubits:
    - $2^{30} \approx 10^{9}$, a billion complex numbers
- For 100 qubits:
    - $2^{100} \approx 10^{30}$, a nonillion complex numbers
- For 300 qubits,
    - $2^{300} \approx 10^{90}$, a Novemvigintillion?

We need more bits than the number of atoms in the universe ($\approx 10^{80}$).



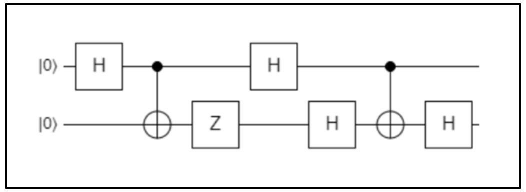
$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$



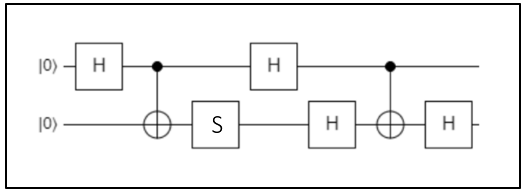
Superposition and Interference

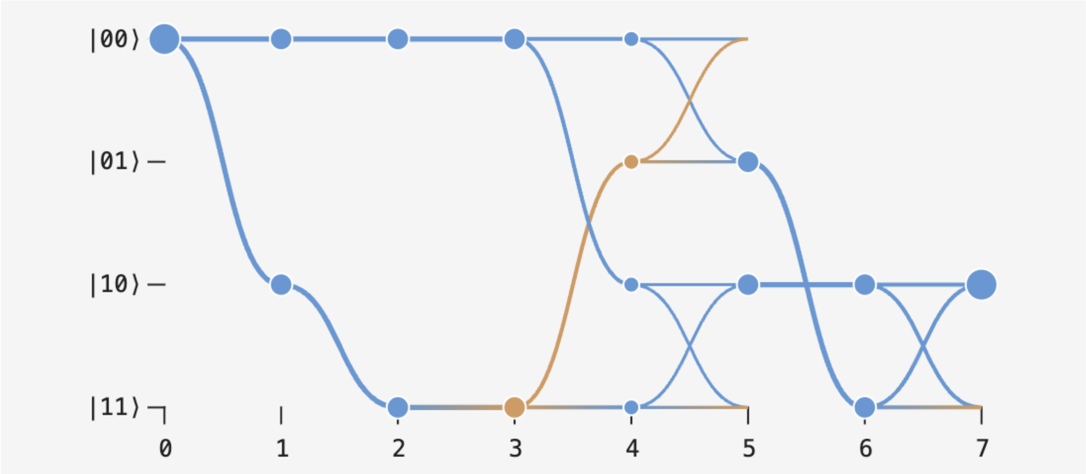
The massive quantum parallelism

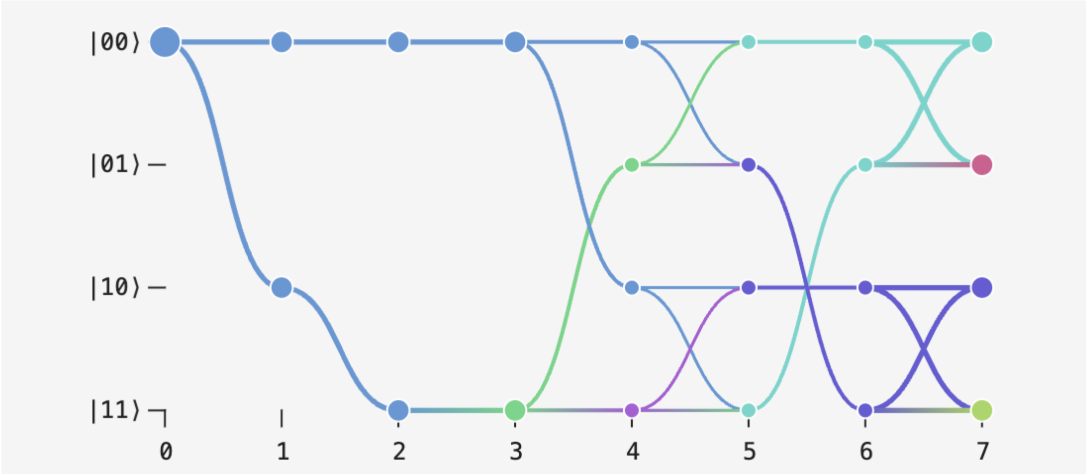
Two-qubit circuit:



Two-qubit circuit:

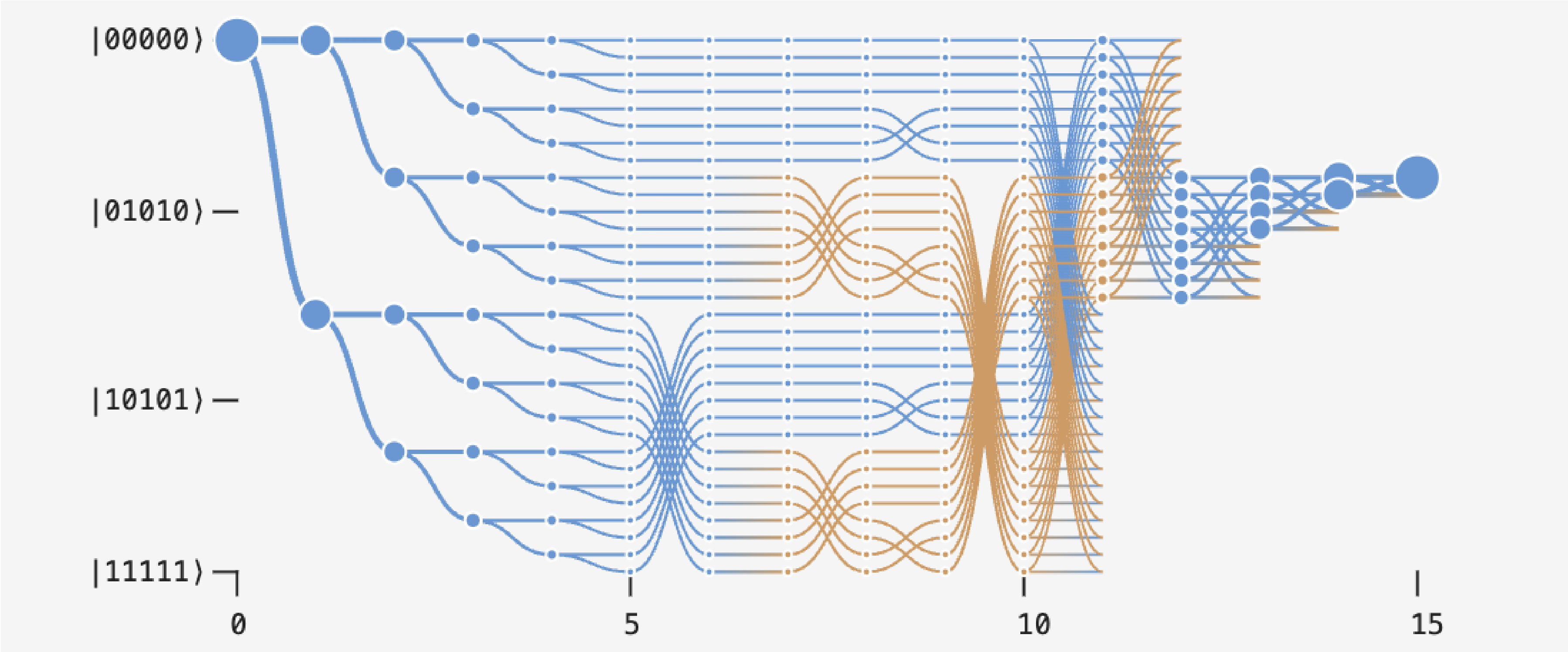






Superposition and Interference

The massive quantum parallelism, and collapse to a high-probability correct output.

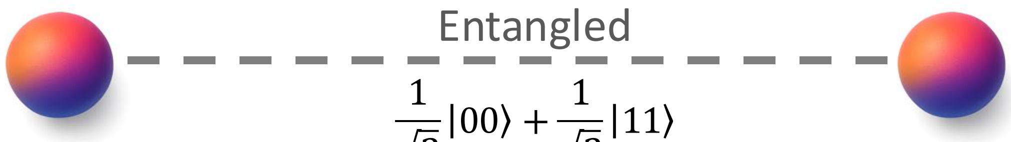


With the right design, quantum interference causes the probability amplitudes of wrong answers to destructively cancel, while those of the right answers constructively amplify.

Entanglement – non-local information

Shared state over a distance

Entangled Quantum Systems



$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

If both qubits measured in the 0/1 basis:

- their outcomes will always be the same.

If measure any one qubit and ignore the other:

- The outcome is a fair coin flip.

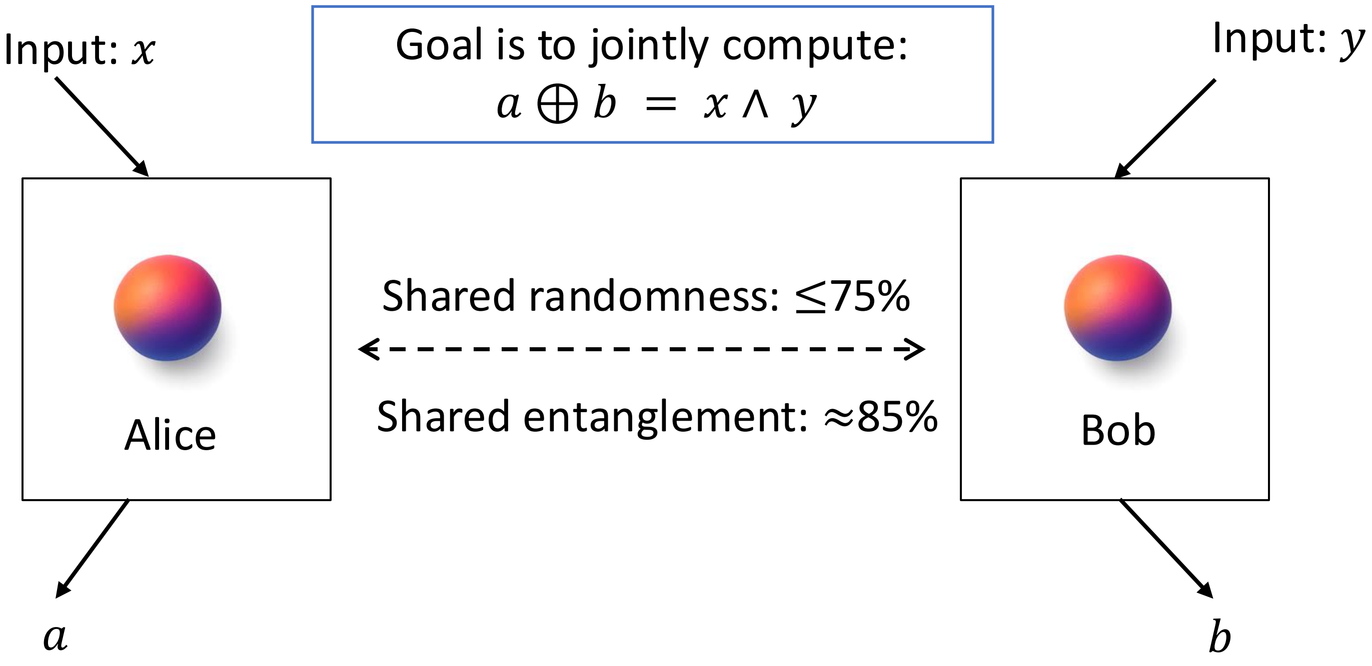
Information is not stored in the individual qubits,
but as "correlations" among the constituent subsystems.

Entanglement as a resource:

- Error-correcting code.
- Distributed computing.
- Certifiable randomness.
- Secure communication.
- Precise quantum sensors.
- Etc.

Entanglement is stronger than classical correlation

Clauser–Horne–Shimony–Holt (CHSH) Game



Theory: John Bell (1964)
Experiment: early ’80s

The Nobel Prize in Physics 2022
Alain Aspect, John F. Clauser, Anton Zeilinger

“for experiments with entangled photons, establishing the violation of Bell inequalities and pioneering quantum information science”

CPSC 4470/5470: Introduction to Quantum Computing

Quantum computational thinking – *how to use superposition and entanglement to solve problems.*

Instructor: Prof. Yongshan Ding (yongshan.ding@yale.edu)

Course Website: https://www.yongshanding.com/cpsc447-f25/

Course Staff:

- Rohan Kumar (rohan.s.kumar@yale.edu)
- Victor Zhou (v.zhou@yale.edu)
- Kun Liu (kun.liu.kl944@yale.edu)